

## 飞浪脚本零起点入门系列(三)

### MAXScript 算术表达式与赋值表达式

作者：飞浪

声明:本教程为 **CG++** 原创,转载请注明出处,谢谢合作:)

本节关键词:

`+` `-` `*` `/` `^` `as` `=` `+=` `-=` `*=` `/=`

本节跟上节紧密联系,希望没学过程序的朋友把开始这几节一定要弄懂。你要是学过程序,了解一下 MAXScript 的规则吧。

#### 算术表达式

MAXScript 算术表达式与我们小学时学的算术一个道理,都是对数值进行运算如加、减、乘、除等。

**MAXScript** 数学运算符有:

`+` 数学中的**加号**,如 `1+1`, `a+b`

`-` 数学中的**减号**,如 `2-1`, `a-b`

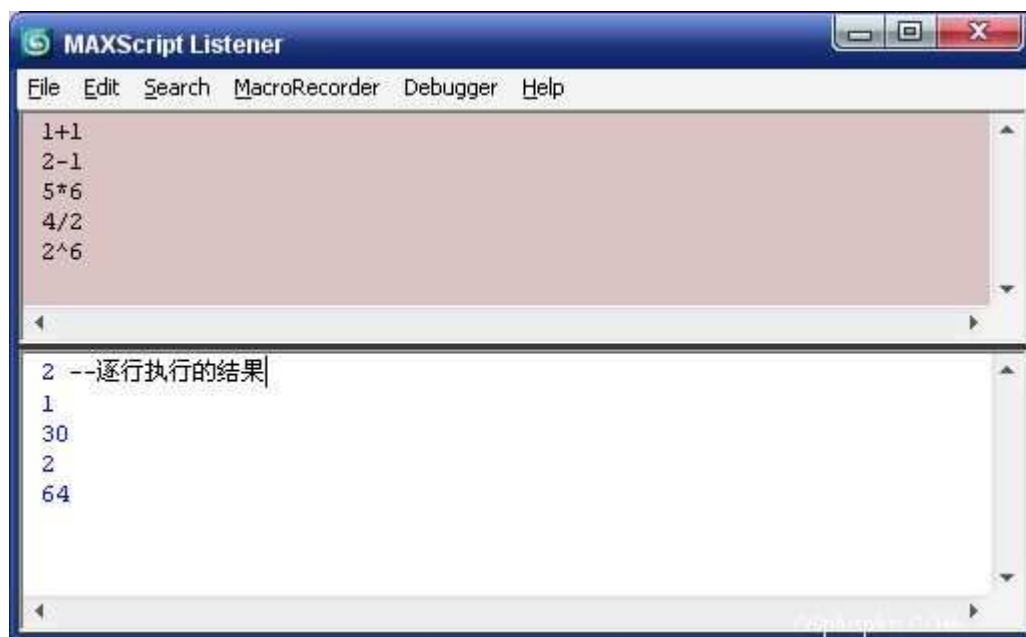
`*` 数学中的**乘号**,如 `5*6`, `a*b`

`/` 数学中的**除号**,如 `4/2`, `a/b`

`^` 数学中的**幂运算**,如 `2^6` 即 2 的 6 次方

`as` **强制类型转换**,严格说这不是数学运算符,以后会发现 `as` 会用在很多地方。

好,现在打开 MAXScript Listener 试试以上数字运算:



运算符可以对数字进行运算，同时运算符左右还可以是函数调用或另外的数学表达式，即复合运算，

如：(a + b) / c; (sin x) \* -d 。

as, 具体表达式为：操作数 as 类型, 这里的类型就包括上节讲的部分数值类型，用法如下：

1.234 as integer --1.234 是 float 类型数值, integer 是整形数值, 这句就是强制把 1.234 变为整形数值, 它不是四舍五入, 而是直接将小数点后面的尾巴去掉, 所以上面那句跟 1.934 as integer 一个效果得到结果 1。

123456789 as string --强制把数值 123456789 转化为字符常量, 结果为"123456789"。

下面大家试试数值转换的结果：



我们知道，数学运算有优先级，以前背的口诀：先算乘除，后算加减，有括号先算括号，那么在 MAXScript 里面也是一样的，唯一不同的是，MAXScript 运算里面只有小括号(), 那么如果我想括号里面再加括号怎么办呢？还是小括号。MAXScript 会自动识别括号的级别，从最里层开始执行。

而中括号[], 大括号{}则另有用途。数学运算的优先级从高到低如下：

操作数

函数调用 --如 sin x,cos y

as

^ --右结合性

\* 和 / --左结合性

+ 和 - --左结合性

混合运算 我们来练习一下上面的运算符号：



`sin 45 * 5`

`(sin 45 * 5) as integer`

`3^8 / "100" as float` --运算顺序为：先进行 `3^8` 运算得到 6561,后面 `as` 运算级别高于 `/`, 所以再 进行 `"100" as float` 运算得 100.0, 最后才进行 `/` 运算即 `6561/100.0` 得 65.61

`2 * pi * 8^2` --`pi` 是常量,即圆周率  $\pi$ ,这句就是计算半径为 8 的圆的面积

`2 * -9 / e` --`e` 常量,即自然对数底

### 赋值表达式

先介绍一下变量,在运行程序的时候,值可以改变的量称为变量。一个变量应该有一个名字,即变量名,如果它的值改变了,你一喊它的名字就知道它作了什么改变了。给变量命名也要遵循一定规则,与其它程序一样,变量名只能由字母、数字和下划线 3 种字符组成,而且第一个字符必须为字母或下划线,而 MAXScript 保留一些关键词是不能用作变量名,如我们上节讲的 `float`,`integer` 等等,因为他们已经被赋予了特殊含义。以下都是合法的变量名:

`cgplusplus` , `sum` , `temp` , `box01` , `btn_rename` , `mat_1` , `RMB10W`

虽然变量名可以随便取,但是建议还是取有一定含义的变量名,以后看到这个名字就知道,哦原来它是代表这个意思,比如 `box01`,一看就知道它指的是 `box`,除非你一定要把一个 `sphere` 赋给它。

MAXScript 变量名不区分大小写,即 `cgpp` 与 `CGpp` 是同一个变量名。

而以下不是合法变量名:

`CG++` , `MR.feilang` , `$1000` , `Yao ming` , `zh-cn` , `cgplusplus.com` , `520HEHE`

赋值符号 `=` 就是赋值运算符,它的作用是将一个数据赋给一个变量。如 `a = 1` 就是执行一次赋值操作,或称赋值运算,把常量 1 赋给 `a`,也可以将一个表达式赋给变量。注意,当你接触了程序你就要知道, `=` 是赋值运算符而不再是数学运算中的等于,程序中的等于符号是 `==` 即两个等于符号。

如下都是对变量进行赋值:



`s = 1`

`sum = ( 1 + 2 + 3 ) * 100 --赋值同时进行了运算`

`mystring = "welcome to cgplusplus.com!"`

MAXScript 提供与 C 语言一样的**复合赋值**，这样提高了编程速度，简化程序。我们先看这样一个赋值：`x = x + 1` 就是将 `x` 的值增加 1，如果之前你赋给 `x = 1`，那么执行此语句后，`x` 就变成了 2，这样对于变量自己的增减操作特别方便，另如：`s = s * 25` 将 `s` 的原来的值乘以 25 后再赋给 `s`，即现在的 `s` 是执行此语句之前的 25 倍。在以后程序中会经常用到这样的操作，程序也给我们提供了一个更简便的写法：`x += 1`，等同于 `x = x + 1`，`s *= 25` 等同于 `s = s * 25`。四个基本数学运算符 `+`，`-`，`*`，`/` 均可与 `=` 结合起来，组成复合赋值符号，例如：

`a += 3` 等同于 `a = a + 3`

`b -= 9` 等同于 `b = b - 9`

`x *= y + 5` 等同于 `x = x * (y + 5)`

`n /= m - p` 等同于 `n = n / (m - p)`

下面出些题，大家思考一下，然后用 MAXScript Listener 执行一下看看结果。

1. `1.8 * 5 + 6 / 2`

2. `58 / 7 as float * 35`

3. `46 * (9 + 12 * (32 - 9)) / "100" as float`

以下假设先赋值：`a = 15`

4. `a += a`

5. `a -= 8`

6. `a *= 3 + 6`

7. `a /= a + a as float`

注意每次执行后 `a` 的值都进行了改变！

这些东西有点枯燥,但是学 MAXScript 却绕不过,刚接触程序可能有些概念一时难以理解,你也可以在以后的学习中逐渐体会,学过程的也请了解一下 MAXScript 的规则,毕竟它远不及 C++ 那么强大,如果你习惯用 C++ 里面的自加“++”符号,到 MAXScript 就行不通了,它不认,呵呵~~~不过 MAXScript 也有自己可爱的一面,比如美元符号\$的使用就很奇妙!

下一节我们来点放松的, MAXScript 对物体的操作,看怎么用 MAXScript 一个简单的语句操作 MAX 物体,请继续关注!

本节原帖地址: <http://www.cgplusplus.com/bbs/viewthread.php?tid=55>

本节结束。